Appendix D
Coordinate Geometry Software

D-1. General

COGO (**CO**ordinate **G**e**O**metry) was initially developed by Charles L. Miller of M.I.T. in 1959. Since then, many improvements have been made, but the basic concept and vocabulary have remained the same. COGO is a problem oriented system that enables the user with limited computer experience to solve common surveying problems. The language is based on familiar surveying terminology, such as, Azimuth, Inverse, Bearing, etc. This terminology is used to define the problem and generate a solution. COGO may be used to solve problems such as curve alignments, point offsets, distance and direction between two points, intersections, etc.

a. The basis of the system is a series of commands used to manipulate or compute points defined by a point number, x-coordinate, and y-coordinate on a plane surface. These points are stored in what is referred to as the "coordinate table" and may be recalled by their point number in future computations.

b. The mathematics used for the computations described in this appendix are beyond the scope of this manual. There are many books published that describe the mathematical procedures in detail.

c. There are many COGO packages on the market today. Several are available to the USACE free of charge. Among these are:

- U0002, by Waterways Experiment Station.
- MCOGO, by Simple Survey Software Inc.
- BLM-COGO, produced by Bureau of Land Management.

All of these are available from the Topographic Engineering Center.

D-2. Requirements

Requirements for COGO are as follows:

a. The ability to utilize a *combined scale factor* in its computations. This will allow the user to calculate the ground distances when staking out a job, or reduce the measured distances to the reference vertical datum, and correct for the scale factor when the survey is to be tied to the SPCS.

b. The ability to rotate and scale (transform) the survey points to fit existing control. When the surveyor uses field coordinates to perform the survey job, the survey can be transformed onto the SPCS by defining two points with their SPCS coordinates.

c. Compass traverse adjustment is sufficient for the majority of traverses established by the USACE. Ability to perform a least squares adjustment can be more advantageous.

d. Must have the ability to work in bearings, north azimuth, or south azimuth.

e. Allow the export of the coordinate table to an ASCII file.

f. Allow the import of points from an ASCII file. Typically, this file will be the ASCII coordinate file produced by total stations and their associated software.

D-3. Functions

COGO functions can be grouped into many categories.

a. Forward Computation Commands used to calculate the coordinates for a point, given the coordinates of a known point and the distance and direction to the unknown point.

(1) LOCATE/AZIMUTH: Computes a point given an azimuth and direction from a known point.

(2) LOCATE/BEARING: Computes a point given a bearing and direction from a known point.

(3) LOCATE/ANGLE: Computes a point given a backsight point, angle, and distance.

(4) LOCATE/LINE: Computes a P.O.T. (point on tangent) given tangent end points and a distance.

(5) LOCATE/DEFLECTION: Computes a point given a backsight, deflection angle, and a distance.

b. Inverse Computation Commands used to compute the distance and direction between two known points. Both the ground and grid distances should be given as output.

(1) INVERSE/AZIMUTH: Computes the distance and azimuth between two known points.

(2) INVERSE/BEARING: Computes the distance and azimuth between two known points.

(3) TANGENT/OFFSET: Computes the distance offline and the distance downline given a known point and the ends of a known tangent.

c. Intersection Commands used to calculate the coordinates of an unoccupied point as the intersection of two vectors of defined direction and/or distance from two known points.

(1) LINE/LINE INT: Computes the coordinates of the point of intersection of two lines whose end points are known.

(2) RANGE/RANGE INT: Computes the coordinates of the intersection of two arcs with known radii and centers. Two answers are possible, so the user must define the desired intersection.

(3) RANGE/AZIMUTH INT: Computes the coordinates of the intersection of a defined vector and an arc. Two answers are possible, so the user must define the desired intersection.

(4) AZIMUTH/AZIMUTH INT: Computes the coordinates of the intersection of vectors with known direction.

(5) FORESECTION: This is an Azimuth/Azimuth intersection, measured by turning angles from two known points.

d. Curve Commands allow the user to define curve parameters to use defined alignment in computations.

(1) ALIGNMENT: Given measured curve parameters, computes components of a curve such as:

- Arc length
- Long chord
- Radius
- Degree of curve
- Tangent length
- Center point coordinates
- External distance
- Mid ordinate
- Central angle

(2) STATION/OFFSET: Computes the coordinates of an unknown point, given a station and offset along the curve. The reverse function is also available to compute the station and offset of a known point relative to the curve alignment.

e. Area Computation Commands which calculate the area of polygons and curve segments. The COGO package should calculate the area based on ground distances, not the reduced distances. This is done by multiplying the computed area by the combined scale factor squared.